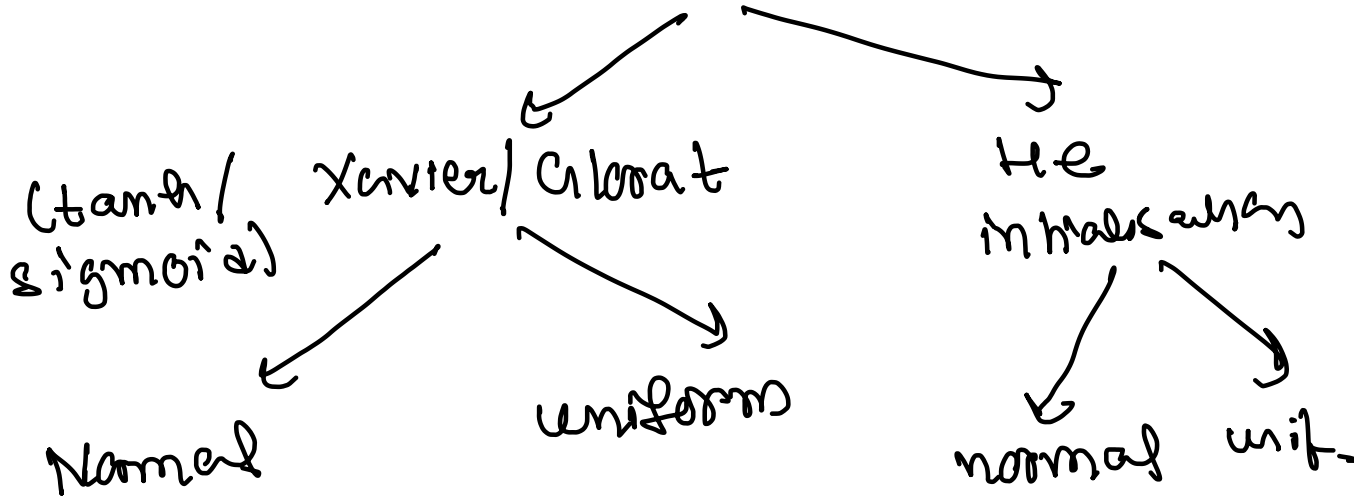
Things not to do—

1) Zero initialization
2) Non-zero constant initialization
3) Random initialization with small weights
4) Random initialization with large weights

what can be done?

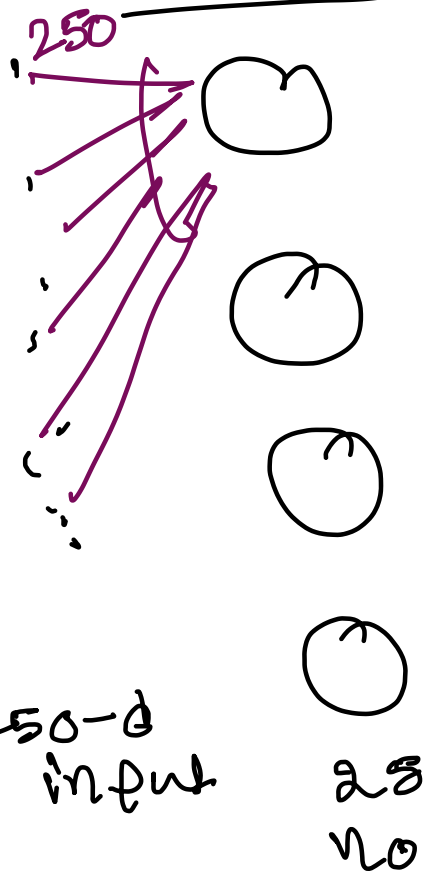
→ Heuristics (Jugad) → practical solution.



* It is proven experimentally when using tanh activation use Xavier / Glorot initialization
* moreover when using Relu activation functions using He initialization.

* variance of randomly assigned values can be very small or very large.

Intuition



small

np.random.randn(250, 250) * 0.01

large

np.random.randn(250, 250) * 1

∴ Variance of the weight should be

$\propto \frac{1}{n}$

∴ where n is the number of input from the previous layer to the nodes.

∴ In the 2nd layer n will 250.

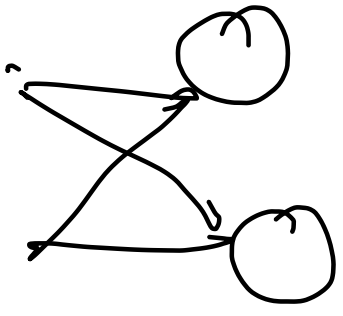
S.D $\frac{1}{\sqrt{n}}$

$z = \sum w_i x_i$

$\eta \rightarrow \wedge \quad \searrow \frac{1}{n}$

Xavier init (Normal) → $\sqrt{\frac{1}{fan\_in}}$ or $\sqrt{\frac{2}{fan\_in + fan\_out}}$

No of inputs coming to nodes



np.random.randn(2,2) * $\frac{1}{\sqrt{2}}$

He Normal

$\sqrt{\frac{2}{fan\_in}}$ ← relu

Uniform Distribution

X_ values → $[-limit, limit]$

$limit = \sqrt{\frac{6}{(fan\_in) + fan\_out}}$

He Uniform

$[-limit, limit]$

$limit = \sqrt{\frac{6}{fan\_in}}$